Статистические методы в биоинформатике

Домашнее задание 1

Правила:

- Дедлайн **17 марта 23:59**.
- Выполненную работу нужно отправить на почту `mipt.stats@yandex.ru`, указав тему письма "`[smb] Фамилия Имя - задание 1`". Квадратные скобки обязательны. Если письмо дошло, придет ответ от автоответчика.
- Прислать нужно ноутбук и его pdf-версию (без архивов).
- Для выполнения задания используйте этот ноутбук в качестве основы, ничего не удаляя из него.
- Никакой код из данного задания при проверке запускаться не будет.

Теоретическая часть (15 баллов)

Каждая задача стоит **5 баллов**.

Задача 1. Дана выборка $X_1, ..., X_n$ из нормального распределения $\mathcal{N}(a, \sigma^2)$. Найдите оценку максимального правдоподобия параметра $\theta = (a, \sigma)$.

Задача 2. Дана выборка $X_1, ..., X_n$ из категориального распределения, то есть распределение на множестве $\{a_1, ..., a_k\}$, причем $\mathsf{P}(X_i = a_j) = p_j$. Найдите оценку максимального правдоподобия параметра $\theta = (p_1, ..., p_k)$.

Задача 3. Дана выборка $X_1, ..., X_n$ из пуассоновского распределения $Pois(\theta)$, то есть $\mathsf{P}(X_i = k) = \frac{\theta^k}{k!}e^{-\lambda}$. Найдите оценку максимального правдоподобия параметра θ.

Задача 1

$X = (X_1, \ldots, X_n)$

$p_i(X, \theta) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x_i-a)^2}{2\sigma^2}}$

$L(X, \theta) = \prod\limits_{i=1}^n p_i(\theta) = \prod\limits_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x_i-a)^2}{2\sigma^2}}$

$\ln L(X, \theta) = Const - n\ln\sigma - \sum\limits_{i=1}^n \frac{(x_i - a)^2}{2\sigma^2}$ - непр. диф. по a, $\sigma > 0$

$\frac{\partial\ln L(X, \theta)}{\partial a} = \sum\limits_{i=1}^n\frac{2(x_i-a)}{2\sigma^2} = \frac{-na + \sum\limits_{i=1}^n x_i}{\sigma^2} = 0 \Rightarrow a = \overline{x}$

$\frac{\partial\ln L(X, \theta)}{\partial \sigma} = -\frac{n}{\sigma} + \frac{\sum\limits_{i=1}^n (x_i - a)^2}{\sigma^3} = 0 \Rightarrow \sigma^2 = \frac{\sum\limits_{i=1}^n (x_i - a)^2}{n} = \overline{(x-a)^2} = \overline{(x-\overline{x})^2} = \overline{x^2-2x\overline{x}+\overline{x}^2} = \overline{x^2} - \overline{x}^2$, т. к. $\overline{x\overline{x}} = \overline{x}^2$

Проверим полож. определенность гессиана:

$\frac{\partial^2\ln L(X, \theta)}{\partial a^2} = -\frac{n}{\sigma^2} < 0$

$\frac{\partial^2\ln L(X, \theta)}{\partial a \partial\sigma} = 0$ при выбранной a

$\frac{\partial^2\ln L(X, \theta)}{\partial \sigma^2} = -2n\sigma^2 < 0$

То есть $\hat{a} = \overline{x}$, $\hat{\sigma} = \sqrt{\overline{(x-\overline{x})^2}} = \sqrt{\overline{x^2} - \overline{x}^2}$ - действительно оценки максимального правдоподобия

Задача 2

$\sum\limits_{j=1}^k p_j = 1$

$L(X, \theta) = \prod\limits_{i=1}^n \prod\limits_{j=1}^k p_j^{I\{x_i = a_j\}} $

$\ln L(X, \theta) = \sum\limits_{i=1}^n \sum\limits_{j=1}^k I\{x_i = a_j\}\ln p_j = \sum\limits_{j=1}^k \#\{x_i|x_i = a_j\}\ln p_j \rightarrow$

$\max\limits_{p_j: \sum\limits_{j=1}^k p_j = 1}, \sum\limits_{j=1}^k \#\{x_i|x_i = a_j\} = n$

Заменим p_k на $1 - \sum\limits_{j=1}^{k-1} p_j$

Для $j=\overline{1,\ldots,k-1} \Rightarrow \frac{\partial\ln L(X, \theta)}{\partial p_j} = \frac{\#\{x_i| x_i = a_j\}}{p_j} - \frac{\#\{x_i| x_i = a_k\}}{1 - \sum\limits_{j=1}^{k-1} p_j} = \frac{\#\{x_i| x_i = a_j\}}{p_j} - \frac{\#\{x_i| x_i = a_k\}}{p_k}$

Чтобы добиться равенства всех этих производных нулю, нужно взять $\hat{p_j} = \frac{\#\{x_i| x_i = a_j\}}{n}$ (других решений нет, т.к. соотношения вероятностей и их общая сумма фиксированы)

Гессиан будет диагонален, на диагонали будут стоять $-\frac{\#\{x_i|x_i = a_j\}}{p_j^2} = -\frac{1}{p_j}$, так что это действительно максимум функции правдоподобия

Задача 3

$L(X, \theta) = \prod\limits_{i=1}^n \frac{\theta^{x_i} e^{-\theta}}{x_k!} = \frac{e^{-n\theta}\theta^{\Sigma x_j}}{x_1!\ldots x_n!}$

$\ln L(X, \theta) = \sum\limits_{i=1}^n x_i \ln\theta - n\theta - \sum\limits_{i=1}^n \ln x_i!$ - непр. диф по θ при $\theta > 0$

$\frac{\partial\ln L(X, \theta)}{\partial\theta} = \frac{\sum\limits_{i=1}^n x_i}{\theta} - n = 0 \Rightarrow \hat{\theta} = \frac{\sum\limits_{i=1}^n x_i}{n} = \overline{x} \ge 0$

$\frac{\partial^2\ln L(X, \theta)}{\partial\theta^2} = -\frac{n^2}{\sum\limits_{i=1}^n x_i }< 0$ (т.к. $x_i \ge 0$)

(вырожденный случай, если все x_i равны 0, но тогда $\ln L(X, \theta) = - n\theta$ и получается тот же ответ, максимальность θ в нуле очевидна)

Практическая часть (30 баллов)

Сначала импортируем некоторые библиотеки. Если некоторые из них не установлены, установите их командой

```
pip install имя_библиотеки
```

Библиотека `Bio` устанавливается с помощью

```
pip install biopython
```

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from Bio import SeqIO
%matplotlib inline
```

Загрузите выданные данные с помощью приведенного ниже кода. Вы будете работать с двумя датасетами. Один из них содержит данные об экспрессии генов с сайта gtexportal.org, Genotype-Tissue Expression -- публичного атласа экспресси генов, содержащим экспрессию генов более чем 50 тканей от 1000 пациентов, а так же их генотип. Другой датасет содержит данные о последовательности энхансеров позвоночных. Тема изучения и предсказания энхансеров, а так же их таргета -- важный аспект современной биоинформатики.

In [2]:

```
gene_counts = pd.read_csv('GTEx_gene_reads.trunc.gct',
                          sep='\t',index_col=0)
records = list(SeqIO.parse("Liver.fasta", "fasta"))
samples_annotation = pd.read_csv('samples_annotation.tsv',
                                 sep='\t')
```

Посмотрим на начало таблиц:

In [3]:

```
gene_counts.head()
```

Out[3]:

	0	1	2	3	4
Name	ENSG00000237613.2	ENSG00000183114.6	ENSG00000060718.14	ENSG00000184599.9	ENSG00000135842.12
Description	FAM138A	FAM43B	COL11A1	FAM19A3	FAM129A
GTEX-1117F-0226-SM-5GZZ7	1	27	652	0	5663
GTEX-111CU-1826-SM-5GZYN	0	43	52	2	1869
GTEX-111FC-0226-SM-5N9B8	1	93	140	4	12268

In [4]:

```
samples_annotation.head()
```

Out[4]:

	SAMPID	SMATSSCR	SMCENTER	SMPTHNTS	SMRIN	SMTS	SMTSD	SMUBRID	SMTSISCH	SMTSPAX	...
0	GTEX-1117F-0226-SM-5GZZ7	0.0	B1	2 pieces, ~15% vessel stroma, rep delineated	6.8	Adipose Tissue	Adipose - Subcutaneous	0002190	1214.0	1125.0	...
1	GTEX-1117F-0426-SM-5EGHI	0.0	B1	2 pieces, !5% fibrous connective tissue, delin...	7.1	Muscle	Muscle - Skeletal	0011907	1220.0	1119.0	...
2	GTEX-1117F-0526-SM-5EGHJ	0.0	B1	2 pieces, clean, Monckebeg medial sclerosis, r...	8.0	Blood Vessel	Artery - Tibial	0007610	1221.0	1120.0	...
3	GTEX-1117F-0626-SM-5N9CS	1.0	B1	2 pieces, up to 4mm aderent fat/nerve/vessel, ...	6.9	Blood Vessel	Artery - Coronary	0001621	1243.0	1098.0	...
4	GTEX-1117F-0726-SM-5GIEN	1.0	B1	2 pieces, no abnormalities	6.3	Heart	Heart - Atrial Appendage	0006631	1244.0	1097.0	...

5 rows × 63 columns

0. С помощью функции `plt.hist` постройте общую гистограмму экспрессии для гена FAM129A из таблицы `gene_counts` для всех `SAMPID`, представленных в `samples_annotation`. Для этого выберите нужные строчки (в данном случаем можно по индексу образцов `SAMPID`) и столбец, где `Description==FAM129A`.

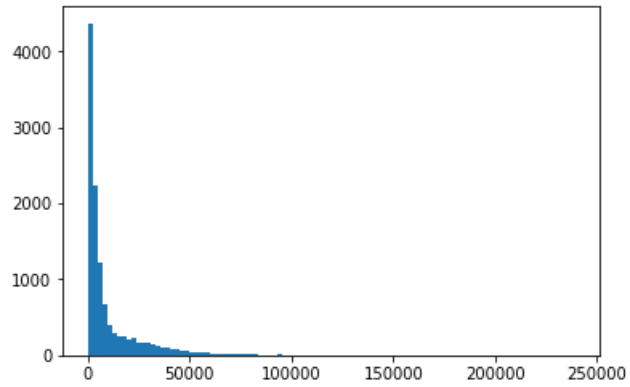
На какое распределение похожа гистограмма?

In [5]:

```
a = gene_counts.loc["Description"] == "FAM129A"
ind = a.index[a == True].tolist()
h = []
for sampid in samples_annotation["SAMPID"]:
    h.append(gene_counts[ind].loc[sampid][0])
results = list(map(float, h))
```

In [6]:

```
plt.hist(results, bins=100)
plt.show()
```

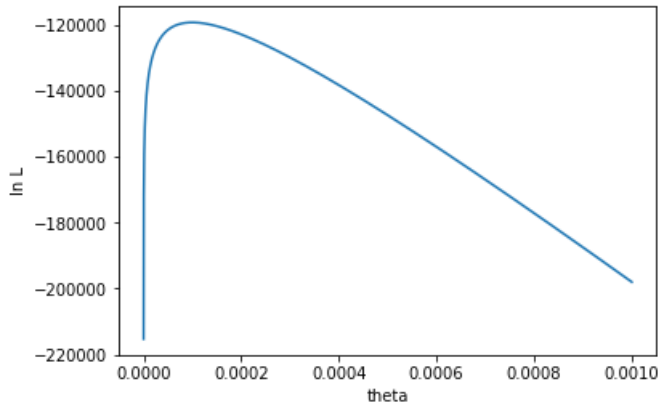


Распределение похоже на экспоненциальное, $p(x, \theta) = \theta e^{-\theta x}, \ln L = n\ln\theta-\theta n \overline{x}$

1. Скорее всего у вас получилось некоторое распределение, параметризованное параметром θ. Найдите оценку максимального правдоподобия этого параметра, построив график логарифмической функции правдоподобия и взяв по нему точку максимума. Чему она равна? Решите данную задачу теоретически и сравните ответ.

In [47]:

```
n = len(results)
x_ = np.mean(results)
start = 1.0e-8
end = 1.0e-3
steps = 100000
x = np.linspace(start, end, steps)
dx = (end-start)/steps
y = []
for el in x:
    s = n*np.log(el)-el*n*x_
    y.append(s)
plt.plot(x, y)
plt.xlabel("theta")
plt.ylabel("ln L")
plt.show()
```



In [29]:

```
i = np.argmax(y)
print("theta = " + str(x[i]) + " +- " + str(dx))
```

```
theta = 9.965999999999999e-05 +- 9.9999e-09
```

$L(x, \theta) = \prod\limits_{i=1}^n \theta e^{-\theta x_i}$

$\ln L(x, \theta) = n \ln\theta -\theta\sum\limits_{i=1}^n x_i$

$\frac{\partial \ln L}{\partial x} = \frac{n}{\theta} - \sum\limits_{i=1}^n x_i$

$\hat{\theta} = \frac{1}{\overline{x}}$

Найдем эту оценку по выборке

In [23]:

```
1/np.mean(results)
```

Out[23]:

```
9.965639943424018e-05
```

Видим, что оценки совпадают с большой точностью, что может указывать на правильность гипотезы о виде распределения (но не доказывает полностью)

2.1. Посчитайте, сколько различных тканей представлено в колонке `SMTSD` таблицы `samples_annotation`. Сколько образцов из артерии большеберцовой кости (`Artery - Tibial`) и легкого (`Lung`)? При выполнении задания может помочь метод `value_counts()`, вызванный у столбца таблицы.

In [42]:

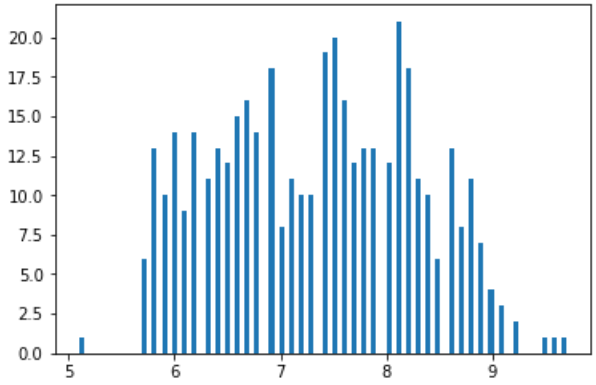
```
print("Totally tissues " + str(len(samples_annotation["SMTSD"].value_counts())))
print("Artery - Tibial " + str(samples_annotation["SMTSD"].value_counts().loc["Artery - Tibial"]))
print("Lung " + str(samples_annotation["SMTSD"].value_counts().loc["Lung"]))
```

```
Totally tissues 53
Artery - Tibial 441
Lung 427
```

2.2. Постройте гистограммы отдельно для артерии и легкого. Что можно о них сказать?

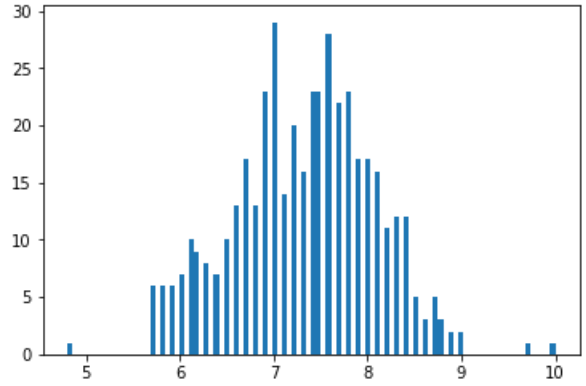
In [73]:

```
lung = samples_annotation["SMTSD"] == "Lung"
ind_lung = lung.index[lung == True].tolist()
data_lung = []
for el in ind_lung:
    data_lung.append(samples_annotation.iloc[el, int(ind[0])])
plt.hist(data_lung, bins=100)
plt.show()
```



In [74]:

```
art = samples_annotation["SMTSD"] == "Artery - Tibial"
ind_art = lung.index[art == True].tolist()
data_art = []
for el in ind_art:
    data_art.append(samples_annotation.iloc[el, int(ind[0])])
plt.hist(data_art, bins=100)
plt.show()
```



Распределения по отдельным органам более похожи на нормальные.

Для таких знаем теоретически, что $\theta = (a, \sigma)$ и $\hat{a} = \overline{x}, \hat{\sigma}^2 = \overline{(x-\overline{x})^2}$

2.3. Посчитайте оценку θ как в пункте 1 отдельно для двух рассматриваемых выше тканей. Что о них можно сказать?

In [75]:

```
mean_lung = np.mean(data_lung)
s = 0
for el in data_lung:
    s += (el - mean_lung)**2
sigma_lung = np.sqrt(s/n)
print("Lung: a = {a}, sigma = {sigma}".format(a=mean_lung, sigma=sigma_lung))

mean_art = np.mean(data_art)
s = 0
for el in data_art:
    s += (el - mean_art)**2
sigma_art = np.sqrt(s/n)
print("Artery: a = {a}, sigma = {sigma}".format(a=mean_art, sigma=sigma_art))
```

```
Lung: a = 7.34028103044, sigma = 0.179002857866
Artery: a = 7.31678004535, sigma = 0.146997042016
```

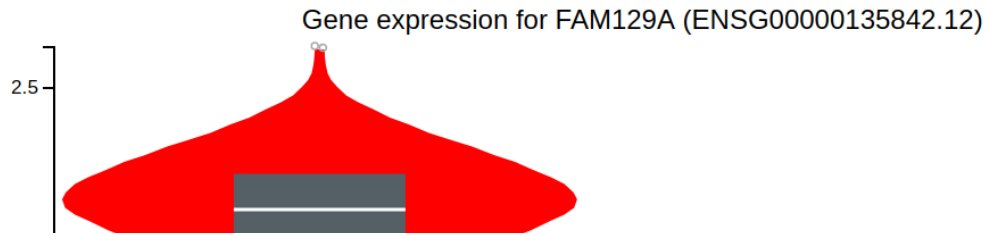
Параметры распределений получились очень похожие, что может говорить о слабой зависимости характера экспрессии этого гена от органа

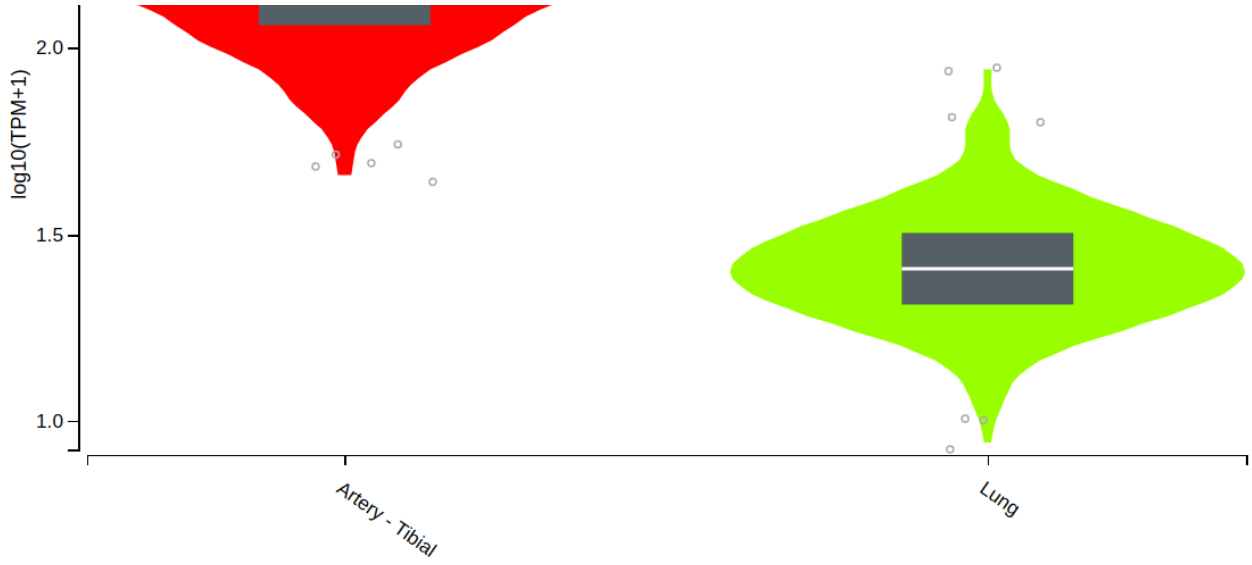
2.4. Зайдите на сайт gtexportal.org и найдите экспрессию гена FAM129A. Правда ли она выше в артерии, чем в легких? В какой ткани наибольшая экспресиия? Доп. вопрос (биология) -- в чем биологическая функция этого гена?

In [81]:

```
from IPython.display import Image
Image("1.png")
```

Out[81]:



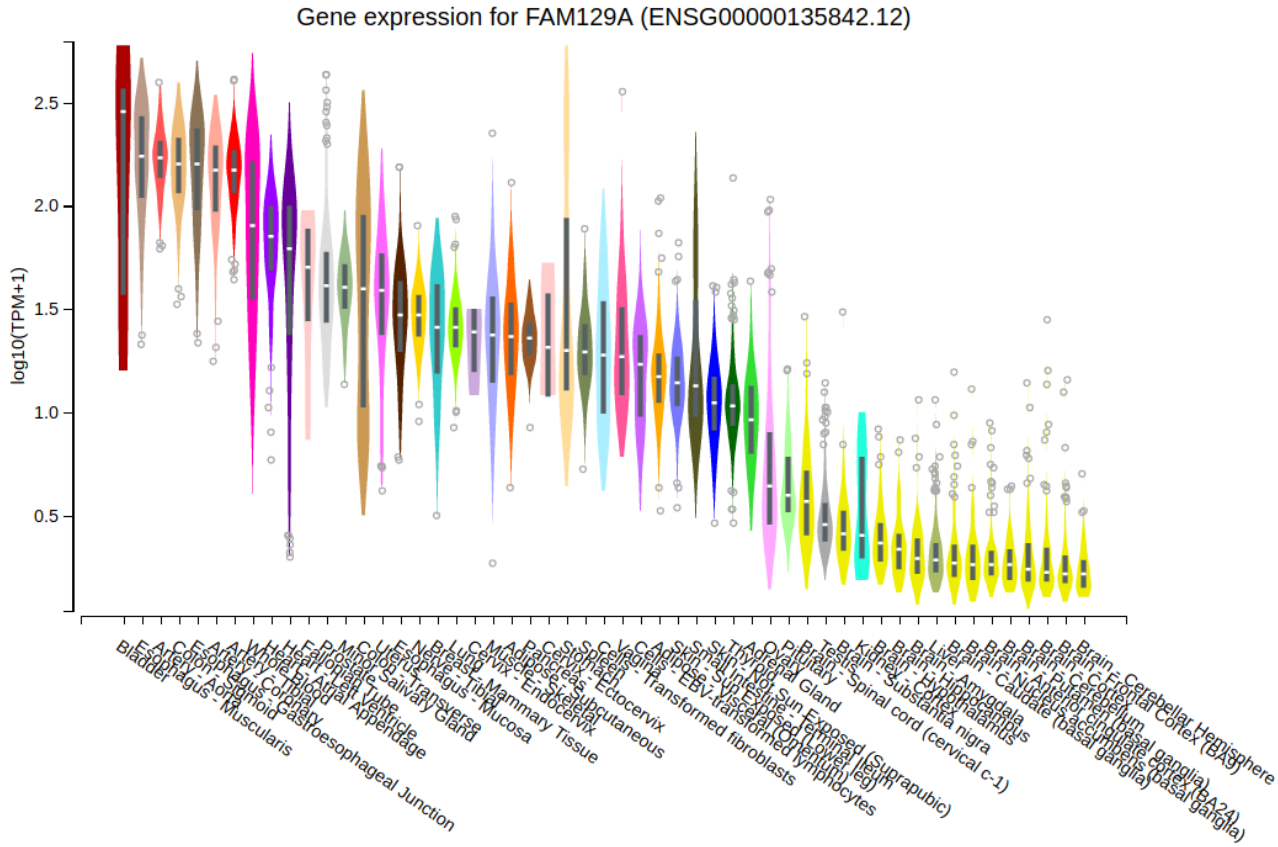


Видно, что в использованной для сайта выборке экспрессия в артерии на полпорядка выше, чем в легких, но есть пересечения по областям (самые нижние в артерии ниже верхних в легких), так что могло оказаться, что данная нам выборка состоит из такого рода данных и потому разницы большой мы не видим

In [80]:

```
Image("2.png")
```

Out[80]:



Наибольшие уровни в мочевом пузыре и пищеводе, хотя в мочевом пузыре большая дисперсия (мало данных)

Экспрессируемый белок влияет на регуляцию трансляции, фосфорилирование и отвечает на эндоплазматический стресс ретикулума http://www.ensembl.org/Homo_sapiens/Gene/Ontologies/biological_process?g=ENSG00000135842;r=1:184790724-184974550

3.0. Посчитайте таблицу встречаемости нуклеотидов первых 10 последовательностей, представленных в `records`, и получив тем самым матрицу четырех числовых последовательностей. Для решения задачи у каждой последовательности

можно вызвать `seq.lower.count('символ')`.

In [135]:

```
N = 10
freq = np.zeros((N,4), dtype="int")
d = {0:"a", 1:"t", 2:"g", 3:"c"}#ATGC
k = 0
for s in records[0:N]:
    for i in range(4):
        freq[k][i] = str(s.seq).lower().count(d[i])
    k += 1
print(freq)
```

```
[[ 431  436  708  715]
 [ 532  399  501  478]
 [ 380  318  452  470]
 [ 303  282  657  558]
 [ 290  302  554  574]
 [1412 1510 2746 2632]
 [1124  929 1330 1317]
 [ 338  412  728  882]
 [ 200  332  274  264]
 [ 807  874  896  933]]
```

Эта функция вам понадобится ниже

In [122]:

```
def stat(obsvd, exptd):
    return ((obsvd - exptd)**2 / exptd).sum()
```

3.1. Посчитайте оценку максимального правдоподобия каждого нуклеотида. Помочь в этом может задача 2 из теоретической части задания. Вы должны получить 4 числа.

In [144]:

```
est = np.zeros(4, dtype="float")
for i in range(4):
    est[i] = np.sum(freq[:,i])/float(np.sum(freq))
print("ATGC")
print(est)
```

```
ATGC
[0.19866803 0.19788251 0.30211749 0.30133197]
```

3.2. Посчитайте ожидаемое количество (математическое ожидание) каждого нуклеотида при данных вероятностях.

In [151]:

```
avg = est*np.sum(freq)
print("For all sequences \n{a}".format(a=avg))
avg_all = np.zeros((N,4), dtype="float")
for i in range(N):
    for j in range(4):
        avg_all[i,j] = est[j] * np.sum(freq[i,:])
print("Every sequence\n{a}".format(a=avg_all))
```

```
For all sequences
[5817. 5794. 8846. 8823.]
Every sequence
[[ 454.94979508  453.15095628  691.84904372  690.05020492]
 [ 379.45594262  377.95560109  577.04439891  575.54405738]
 [ 321.84221311  320.56967213  489.43032787  488.15778689]
 [ 357.60245902  356.18852459  543.81147541  542.39754098]
 [ 341.70901639  340.3579235   519.6420765   518.29098361]
 [1648.94467213 1642.42486339 2507.57513661 2501.05532787]
 [ 933.7397541   930.04781421 1419.95218579 1416.2602459 ]
 [ 468.85655738  467.00273224  712.99726776  711.14344262]
 [ 212.57479508  211.73428962  323.26571038  322.42520492]
```

```
[ 697.32479508  694.56762295 1060.43237705 1057.67520492]]
```

3.3. Сгенерируйте случайную матрицу 4x10, используя полученный ранее вектор вероятностей оценки максимального правдоподобия. Для генерации воспользуйтесь функцией `scipy.stats.multinomial.rvs`. Вы должны получить матрицу 4x10, причем итоговое число нуклеотидов для каждой сгенерированной последовательности должны быть равно их изначальной длине.

In [178]:

```
from scipy.stats import multinomial

total = np.zeros(N, dtype="float")
sluch = np.zeros((N, 4), dtype="float")
for i in range(10):
    total[i] = np.sum(freq[i,:])
    sluch[i] = multinomial.rvs(total[i],est)
print(sluch)
```

```
[[ 461.  458.  669.  702.]
 [ 376.  402.  584.  548.]
 [ 315.  305.  497.  503.]
 [ 374.  395.  505.  526.]
 [ 346.  341.  537.  496.]
 [1669. 1616. 2544. 2471.]
 [ 953.  928. 1414. 1405.]
 [ 454.  471.  707.  728.]
 [ 187.  214.  342.  327.]
 [ 745.  670. 1059. 1036.]]
```

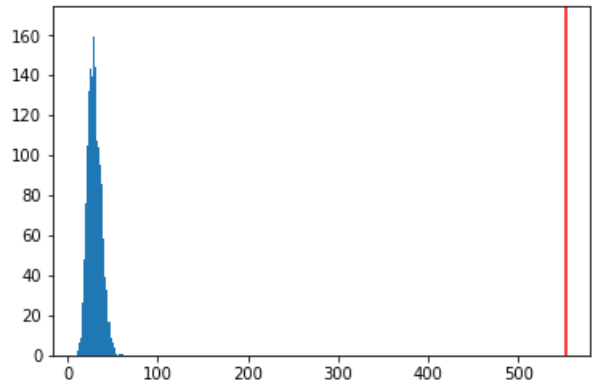
3.4. Сгенерируйте такую матрицу 5000 раз. Для каждой итерации посчитайте статистику между ожидаемым и сгенерированным с помощью функции `stat()`. Постройте гистограмму и отметьте значение этой статистики для исходных последовательностей. Что можно сказать? Можете ли вы указать связь с проверкой статистических гипотез?

In [181]:

```
st = []
for j in range(5000):
    for i in range(10):
        total[i] = np.sum(freq[i,:])
        sluch[i] = multinomial.rvs(total[i],est)
    st.append(stat(sluch,avg_all))
```

In [194]:

```
inp = stat(sluch,freq)
plt.hist(st, bins=100)
plt.axvline(inp,color='red')
plt.show()
```



Видим, что исходные данные лежат значительно дальше (во многих стандартных отклонениях) от распределения случайных данных, построенных при гипотезе, что распределение нуклеотидов имеет одинаковые вероятности для всех последовательностей.

Таким образом, вероятнее всего эту гипотезу нужно отклонить